
Proceedings Paper:
Dale, Matthew, Stepney, Susan orcid.org/0000-0003-3146-5401 and Trefzer, Martin Albrecht orcid.org/0000-0002-6196-6832 (2020) Designing Computational Substrates using Open-Ended Evolution. In: Artificial Life Conference Proceedings. MIT Press , pp. 665-667.

https://doi.org/10.1162/isal_a_00294

Designing Computational Substrates using Open-Ended Evolution

Matthew Dale, Susan Stepney and Martin Trefzer

University of York
matt.dale@york.ac.uk

Abstract

Evolutionary algorithms are powerful tools to discover novel and diverse solutions to complex problems. Here, we discuss how open-ended algorithms, such as novelty search, can be used to design and evaluate new unconventional computing systems, from the design of materials to the creation of new computational models.

Introduction

Computing with unconventional materials is a growing area of research across many disciplines (Adamatzky, 2016a,b). Unconventional computers have the potential to be faster and consume less power than conventional CMOS technology (Stepney et al., 2018). However, to achieve this requires innovations in material design and the discovery of new computational models.

Designing new materials creates significant technical challenges often requiring advanced computer modelling, new fabrication techniques, ingenuity and creativity. There are many approaches to material design, for example, bottom-up design where structures can grow and self-organise, or top-down design where 'basic' components are connected together in a well-defined manner. Design and validation of any material is typically expensive, labour intensive and requires considerable expertise.

The other challenge in unconventional computing is designing new computational models. Getting the right computational model is critical. The computational model should naturally fit the material's implementation; a poor model causes inefficiencies, requires extensive engineering and may ignore promising qualities of the material.

Computational models are typically abstracted from or inspired by the behaviours of specific physical or biological processes. One problem is that designing new models is difficult without some formal language to express them. Another problem is how to experimentally validate these models, how to assess their suitability for a specific material.

To reduce the need for expertise and to automate design, search algorithms inspired by natural evolution are often used. Artificial evolution requires minimal prior knowledge of the system; this makes them easy to implement and removes designer bias. In the field of Evolvable Hardware, evolution is used to create and optimise the design of hardware systems, from analogue circuits to robots (Tan et al., 2004; Doncieux et al., 2015). Recently, algorithms inspired by the qualities of open-ended evolution, such as novelty search and quality diversity algorithms (Lehman and Stanley, 2008; Pugh et al., 2016), have added new methods to design artefacts, producing solutions that direct optimisation often struggles to recreate.

We have developed a framework (Dale et al., 2019b) that exploits novelty search to assess the quality of material substrates. Here we discuss how it can be modified to improve material design and to evaluate new computational models.

Computing with Materials

To compute with materials we need to be able to determine when some desired abstract computation is performed, in contrast to the material undergoing other natural physical processes. To determine when a physical or biological system is computing, abstraction/representation theory (AR theory) has been developed (Horsman et al., 2014, 2017; Stepney and Kendon, 2019). AR theory defines when a material substrate is computing with respect to a model, according to a representation. The theory defines the general compute cycle for a physical computer, starting with an initial abstract problem, the encoding step in terms of an abstract computational model, and its instantiation into a physical material. The theory defines the compute process and whether the system fulfils the computing definition, but it does not evaluate the efficiency of the material or the suitability of the model.

We have developed a framework to explore and compare the computational expressiveness and capability of physical materials with respect to one particular computational model, that of Reservoir Computing (Lukoševičius et al., 2012). This CHARC (CHAracterisation of Reservoir Computers) framework (Dale et al., 2019b) provides a means to characterise a material's *quality* for reservoir computing according to its range and diversity of physical behaviours.

Reservoir computing is a popular computational model to exploit a range of novel physical computing devices. Its structure and simplicity make it suitable to be implemented in many open non-linear dynamical systems. A recent review of physical reservoirs highlights the diversity of systems the model is applied to, including chemical, optical, electronic and mechanical systems (Tanaka et al., 2019).

There are typically two stages to program a physical reservoir. The first stage is to find a set of physical configuration parameters that induce desirable system dynamics. In this state, the reservoir acts as a dynamical filter on its inputs. The second stage is to train a separate task-specific readout layer, typically forming a linear combination of system states.

What tailors the CHARC framework to reservoir computing is the dynamical properties and metrics used to define a behaviour. In reservoir computing, some properties are described as being essential, such as non-linearity and a fading memory (Dambre et al., 2012).

CHARC Framework Outline

The CHARC framework measures the *quality* of a material, where quality is defined as the total capacity to realise distinct reservoirs in terms of different dynamical properties. To characterise a test material, two phases must be completed: quality assessment of a *reference* material (phase one), and characterisation of the *test* material (phase two). Phase one provides a baseline to compare to, and is typically carried out only once, provided a suitable reference is chosen. The basic process for each phase has three steps.

Step one, create an abstract space to explore, map, and measure. This space represents the dynamical properties of the material when configured. We refer to this space as the *behaviour space*, inspired by the same representation used in novelty search. To form each behaviour, n independent property measures are used. Increasing the number of measures leads a more detailed representation of the material, but also increases the size of search space – a suitable trade-off is therefore needed. In previous work, three measures are used to define the behaviour space: Kernel Rank (non-linearity), Generalisation Rank (stability), and Memory Capacity; for more information about these measures, see Dale et al. (2019b). These measures are somewhat generic and build a basic dynamical picture of the material. For example, low values in both rank measures signify a material configured in an ordered regime, and high values equate to chaotic regimes.

Step two, explore the material configurations. Here, the mapping between abstract reservoir and material configuration is explored. Exploration is carried out in the behaviour space using novelty search (Lehman and Stanley, 2008). Novelty search, an open-ended and objective-free genetic algorithm, navigates the behaviour space searching for novel solutions. In this implementation, every behaviour considered is stored in an external database for later use. This database forms the core resource used to analyse the relationship between parameters, tasks and behaviours after the search process.

Step three, measure the quality. To do this, the behaviour space is divided into voxels; the number and size of voxels depends on the spaces being compared. The total number of voxels occupied by discovered behaviours forms the measure of quality. The quality value therefore represents an approximation of the system's dynamical freedom, or, the material's capacity to instantiate different reservoirs.

Substrate Design

In Dale et al. (2019b), CHARC is used to manipulate a limited set of parameters referred to as *configuration* parameters. These do not change the physical material, only how to interact with it. In AR theory, this would cover how to encode the abstract problem and instantiate the material.

To explore material *design*, material properties can be added to the parameter search space, for example, parameters that define the physical structure of the material, or the natural unperturbed behaviour of the material. Properties such as these could be realised through fabrication techniques or specific layouts of components.

To demonstrate the concept, CHARC has been used to compare different simulated network topologies of varying complexities as an analogy for material design (Dale et al., 2019a). Using CHARC, the dynamical limitations and boundaries of different structures are characterised. It is shown that simple structures with greater network size can mimic the same behaviours of smaller complex network structures.

Computational Model Design

The CHARC framework is not fundamentally limited to the reservoir computing model. The measures defining the behaviour space are adapted according to the specific computational model. The same process is then repeated as before.

Given a suitable language for computational models, such as a general dynamical systems representation (Stepney, 2019), novelty search could explore the space of models whilst parameters of the material remain constant. A co-design approach is also possible (Stepney, 2019). Both material and model design are combined, to improve each individually and also the fit between them.

We believe that this framework could be exploited for co-designing materials and models for many forms of ALife, such as soft-bodied robots (Cheney et al., 2014) and other forms of embodied cognition.

Acknowledgments. The work reported here was part-funded by a Defence Science and Technology Laboratory (DSTL) PhD studentship, and part-funded by the SpInsired project, EPSRC grant no. EP/R032823/1.

References

Adamatzky, A., editor (2016a). *Advances in Unconventional Computing: Volume 1: Theory*. Springer.

Adamatzky, A., editor (2016b). *Advances in Unconventional Computing: Volume 2 Prototypes, Models and Algorithms*. Springer.

Cheney, N., MacCurdy, R., Clune, J., and Lipson, H. (2014). Unshackling evolution: Evolving soft robots with multiple materials and a powerful generative encoding. *ACM SIGEvolution*, 7(1).

Dale, M., Dewhirst, J., O'Keefe, S., Sebald, A., Stepney, S., and Trefzer, M. A. (2019a). The role of structure and complexity on reservoir computing quality. In *International Conference on Unconventional Computation and Natural Computation*, pages 52–64. Springer.

Dale, M., Miller, J. F., Stepney, S., and Trefzer, M. A. (2019b). A substrate-independent framework to characterize reservoir computers. *Proceedings of the Royal Society A*, 475(2226):20180723.

Dambre, J., Verstraeten, D., Schrauwen, B., and Massar, S. (2012). Information processing capacity of dynamical systems. *Scientific Reports*, 2.

Doncieux, S., Bredeche, N., Mouret, J.-B., and Eiben, A. E. G. (2015). Evolutionary robotics: what, why, and where to. *Frontiers in Robotics and AI*, 2:4.

Horsman, C., Stepney, S., Wagner, R. C., and Kendon, V. (2014). When does a physical system compute? *Proc. Roy. Soc. A*, 470(2169):20140182.

Horsman, D., Stepney, S., and Kendon, V. (2017). The natural science of computation. *Communications of ACM*, 60:31–34.

Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336.

Lukoševičius, M., Jaeger, H., and Schrauwen, B. (2012). Reservoir computing trends. *KI-Künstliche Intelligenz*, 26(4):365–371.

Pugh, J. K., Soros, L. B., and Stanley, K. O. (2016). Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40.

Stepney, S. (2019). Co-designing the computational model and the computing substrate. In *UCNC 2019, Tokyo, Japan, June 2019*, volume 11493 of *LNCS*, pages 5–14. Springer.

Stepney, S. and Kendon, V. (2019). The role of the representational entity in physical computing. In *UCNC 2019, Tokyo, Japan, June 2019*, volume 11493 of *LNCS*, pages 219–231. Springer.

Stepney, S., Rasmussen, S., and Amos, M., editors (2018). *Computational Matter*. Springer.

Tan, K. C., Wang, L., Lee, T. H., and Vadakkepat, P. (2004). Evolvable hardware in evolutionary robotics. *Autonomous Robots*, 16(1):5–21.

Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano, D., and Hirose, A. (2019). Recent advances in physical reservoir computing: A review. *Neural Networks*.